AR

Requested Patent: JP2001007822A

Title:

A PIPELINED PACKET SCHEDULER FOR HIGH SPEED OPTICAL SWITCHES ;

Abstracted Patent: EP1061763 ;

Publication Date: 2000-12-20 ;

Inventor(s): CAVENDISH DIRCEU G (US) ;

Applicant(s): NIPPON ELECTRIC CO (JP) ;

Application Number: EP20000103463 20000229 ;

Priority Number(s): US19990335908 19990618; US19990460649 19991214 ;

IPC Classification: H04Q11/00 ;

Equivalents:

ABSTRACT:

A pipelined scheduler which allows easy implementation and control and further fair scheduling among input lines of a crossbar high speed switch fabric is discussed. By means of a round-robin communication scheme, a systematically ordered sequence of visits to time slots can be obtained regardless of whether the number of scheduler modules is even or odd by framing the time axis and using a priority matrix to reserve future time slots. Further, a Carry Over Round-robin Pipelined Scheduler (CORPS) achieves scalability to a large number of ports. Moreover, CORPS achieves one scheduling decision per line per slot, by scheduling packets in future slots. It is well suited to the support of Quality of Service traffic, since the choice of the queues to be scheduled is arbitrary. CORPS limits itself to resolve, in a fair way, the contention for output ports.

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－7822
（P2001－7822A）
(43)公開日　平成13年１月12日(2001.1.12)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード'（参考）
Ｈ０４Ｌ　12/28		Ｈ０４Ｌ　11/20	Ｈ	
12/56			Ｇ	
			１０２Ｃ	

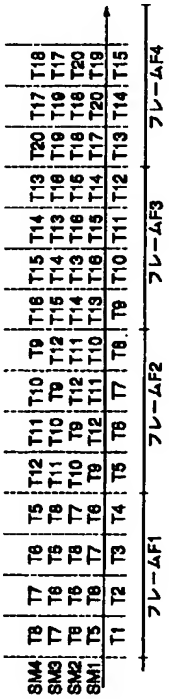
審査請求　有　請求項の数17　ＯＬ　（全 20 頁）

(21)出願番号　特願2000－55103(P2000－55103)
(22)出願日　平成12年３月１日(2000.3.1)
(31)優先権主張番号　０９／３３５９０８
(32)優先日　平成11年６月18日(1999.6.18)
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０９／４６０６４９
(32)優先日　平成11年12月14日(1999.12.14)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(72)発明者　ディアスー・ジー・キャベンディッシュ
アメリカ合衆国、ニュージャージー　08540　プリンストン，４　インディペンデンス　ウエイ、エヌ・イー・シー・ユー・エス・エー・インク内
(74)代理人　100097157
弁理士　桂木　雄二

(54)【発明の名称】　データフロー制御スイッチ及びそのスケジューリング方法

(57)【要約】
【課題】　実装及び制御が容易なパイプラインスケジューラおよびクロスバ高速スイッチファブリックの入線間での公平なスケジューリングが可能なパイプラインスケジューラを提供する。
【解決手段】　時間軸をフレーム化し、将来のタイムスロットを予約するための優先マトリクスを用いたラウンドロビン方式により、スケジュールモジュールが偶数個か奇数個に関係なく、タイムスロットの規則的な巡回順序が得られる。さらに、繰越しラウンドロビンパイプラインスケジューラ（ＣＯＲＰＳ）は多数のポートへのスケーラビリティを実現する。また、ＣＯＲＰＳは、将来のスロットのパケットをスケジューリングすることにより、ラインごとスロットごとに１つのスケジューリング決定を行う。スケジューリングされるキューの選択は任意であるため、本発明は、トラフィックのサービス品質をサポートすることに適している。ＣＯＲＰＳは、出力ポート間の競合を公平に解決する。



【特許請求の範囲】
【請求項１】 ネットワークにおけるデータのフローを制御するスイッチにおいて、
複数の入力ポートと、
複数の出力ポートと、
前記複数の出力ポートのうちの指定出力ポートへデータを送るように、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングする複数の入力ポートスケジュールモジュールを有するスケジューラと、からなり、
現在のスケジュールモジュールは、
前のスケジュールモジュールからスケジューリングメッセージを受信し、
前記現在のスケジュールモジュールが前記指定出力ポートにアクセスしようとする将来のタイムスロットを計算し、
前記将来のタイムスロットが前記現在のスケジュールモジュールによって既に予約されているかどうか、前記将来のタイムスロットが阻止されているかどうか、及び前記将来のタイムスロットが他のスケジュールモジュールによって取られているかどうかに基づいて、前記将来のタイムスロットが有効かどうかを判断し、
有効な場合には、前記将来のタイムスロットを取り、前記スケジューリングメッセージに前記将来のタイムスロットが取られたことを示す情報を入れる、
ことを特徴とするデータフロー制御スイッチ。
【請求項２】 前記スケジューラは、前記将来のタイムスロットが予約されている場合及び取られている場合のいずれかである時には、前記将来のタイムスロットを所定数のタイムスロットだけ前進させることを特徴とする請求項１記載のデータフロー制御スイッチ。
【請求項３】 前記複数の出力ポートのそれぞれに対して別々のキューを維持する仮想出力キューイング（ＶＯＱ）を用いて、前記複数の入力ポートを通じて入力されたデータをキューイングすることを特徴とする請求項１記載のデータフロー制御スイッチ。
【請求項４】 あるポートに対する前記仮想出力キューイングは、他のポートに対する前記仮想出力キューイングとは独立であることを特徴とする請求項３記載のデータフロー制御スイッチ。
【請求項５】 前記仮想出力キューイングのサービスレートは予測可能かつ調整可能であることを特徴とする請求項３に記載のデータフロー制御スイッチ。
【請求項６】 前記スケジューラは、重み付きラウンドロビンに基づいて、前記指定出力ポートを選択することを特徴とする請求項１記載のデータフロー制御スイッチ。
【請求項７】 複数の入力ポートスケジュールモジュールを有するスイッチの複数の入力ポートに到着する入力信号を当該スイッチの複数の出力ポートに送るようにスケジューリングする方法において、
ａ）現在のスケジュールモジュールが、前のスケジュールモジュールからスケジューリングメッセージを受信するステップと、
ｂ）前記現在のスケジュールモジュールが、前記複数の出力ポートのうちの１つにアクセスしようとする将来のタイムスロットを計算するステップと、
ｃ）前記複数の出力ポートのうちの１つを前記将来のタイムスロットでの送信用にスケジューリングするように選択するステップと、
ｄ）前記将来のタイムスロットが前記現在のスケジュールモジュールによって既に予約されているかどうかを判断するステップと、
ｅ）前記将来のタイムスロットが前記現在のスケジュールモジュールによって予約されていない場合には、前記将来のタイムスロットが阻止されているかどうかを判断するステップと、
ｆ）前記将来のタイムスロットが阻止されていない場合には、前記将来のタイムスロットが他のスケジュールモジュールによって既に取られているかどうかを判断するステップと、
ｇ）前記将来のタイムスロットが、他のスケジュールモジュールによって既に取られている場合及び前記現在のスケジュールモジュールによって既に予約されている場合のいずれかの場合には、前記スケジューリングメッセージから繰越し動作が既に開始されているかどうかを判断するステップと、
ｈ）前記繰越し動作が既に開始されている場合には、前記将来のタイムスロットを阻止状態に設定して前記ステップ（ｄ）に戻るステップと、
ｉ）前記繰越し動作が開始されていない場合には、前記将来のタイムスロットを所定数のタイムスロットだけ前進させ、繰越しフラグをセットしてステップ（ｄ）に戻るステップと、
ｊ）前記将来のタイムスロットが他のスケジュールモジュールによって取られていない場合には、前記将来のタイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、
ｋ）前記スケジューリングメッセージを次のスケジュールモジュールに渡すステップと、
からなることを特徴とするスケジューリング方法。
【請求項８】 前記複数の入力ポートを通じて入力したデータは、各出力ポートに対して別々のキューを維持する仮想出力キューイングを用いてキューイングされることを特徴とする請求項７記載の方法。
【請求項９】 あるポートに対する前記仮想出力キューイングは、他のポートに対する前記仮想出力キューイングとは独立であることを特徴とする請求項８記載の方法。

【請求項１０】 前記仮想出力キューイングのサービスレートは予測可能かつ調整可能であることを特徴とする請求項８記載の方法。
【請求項１１】 前記スケジューラは、重み付きラウンドロビンに基づいて、前記指定出力ポートを選択することを特徴とする請求項７記載の方法。
【請求項１２】 ネットワークにおけるデータのフローを制御するスイッチにおいて、
複数の入力ポートと、
複数の出力ポートと、
前記複数の出力ポートのうちの指定出力ポートへデータを送るように、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングするＮ個の複数の入力ポートスケジューリングモジュールを有するスケジューラと、
から構成され、
前記スケジューラは、
前記各入力ポートスケジューリングモジュールがリング状に接続され、
タイムスロット単位に、
各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから、ある予約タイムスロットの予約状況情報を受信し、
各入力ポートスケジューリングモジュールが、その予約タイムスロットにおける当該入力ポートスケジューリングモジュールからのパケット送出予約可否を決定し、
各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから受信した予約状況情報に、自スケジューリングモジュールの予約結果を反映させて、次段のスケジューリングモジュールに送信する、
ことを特徴とするデータフロー制御スイッチ。
【請求項１３】 複数の入力スケジューリングモジュールを有するパケットスイッチの入力ポートと出力ポートの接続状態を決定し接続を予約する（以下、スケジューリングという。）方法において、
Ｎ個のタイムスロットを単位とするフレームを定義して、前記フレーム時間内で、前記フレームの次フレーム中のＮ個のタイムスロットでのスケジューリングを行う、
ことを特徴とするスケジューリング方法。
【請求項１４】 前記スケジューリング方法は、
ａ）現在のスケジュールモジュールが、前のスケジュールモジュールからスケジューリングメッセージを受信するステップと、
ｂ）前記現在のスケジュールモジュールが、前記複数の出力ポートのうちの１つにアクセスしようとする将来のタイムスロットを予め次フレーム内の特定のタイムスロットに決定するステップと、
ｃ）前記複数の出力ポートのうちの１つを前記将来のタイムスロットでの送信用にスケジューリングするように選択するステップと、
ｄ）前記将来のタイムスロットが他のスケジュールモジュールによって既に取られているかどうかを判断するステップと、
ｅ）前記将来のタイムスロットが他のスケジュールモジュールによって取られていない場合には、前記将来のタイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、
ｆ）前記スケジューリングメッセージを次のスケジュールモジュールに渡すステップと、
からなることを特徴とする請求項１３記載のスケジューリング方法。
【請求項１５】 前記スケジューリング方法は、
タイムスロットでのスケジューリング決定過程（接続決定過程）の観点から見た場合、前記複数の接続決定過程が、
フレームの先頭で同時に開始され、
フレーム内で同時にパイプライン処理により進行し、
フレームの末端で同時に完了する、ことを特徴とする請求項１３記載のスケジューリング方法。
【請求項１６】 前記スケジューリング方法は、
前記入力ポートスケジューリングモジュールが、フレームの先頭で同時に開始する前記各接続決定過程において、次のフレーム内の各々異なる予約タイムスロットを対象として処理を開始することを特徴とする請求項１３記載のスケジューリング方法。
【請求項１７】 前記スケジューリング方法は、
ある与えられた将来のタイムスロットにおけるＮ個の入力ポートスケジューリングモジュールの規則的な巡回順序を定義するＮ×Ｎマトリクスを参照することによって、現在のフレームにおける入力信号が次のフレームでどの出力ポートへ送出されるかを決定することを特徴とする請求項１３記載のスケジューリング方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、ネットワークによるデータのフローを制御するネットワークシステムおよびスイッチに係り、特に、大容量スイッチを通じてのデータフローを管理するスケジューラに関する。
【０００２】
【従来の技術】入力キュースイッチアーキテクチャは、常に高速交換システムの有力な選択肢であった。それは主に、入力バッファのメモリアクセス速度が、全交換容量とともにスケールするのではなく、単一の入線の速度とともにスケールするからである。しかし、入力バッファ型スイッチは、ＨＯＬ(head-of-line)ブロッキングの問題があることが以前から知られており、これにより全スループットは５８．６％という理論限界に制限される（M. J. Karol, M. G. Hluchyj, S. P. Morgan,"Input

Versus Output Queuing on a Space-Division Packet Switch", IEEE Transactions on Communications, Vol.COM-35, No.12, pp.1347-1356, Dec. 1987、参照）。

【０００３】最近になって、入力スイッチのＨＯＬブロッキングの問題を克服するために、仮想出力キューイング（ＶＯＱ：Virtual Output Queuing）という入力キューイング方式が提案された（Y. Tamir and G. Frazier, "High Performance Multi-queue Buffers for VLSI Communication Switches", Proceedings of 15th Ann.Symp. on Comp. Arch., pp.343-354, June 1988、および、T. Anderson, S. Owicki, J. Saxe, C. Thacker, "High Speed Switch Scheduling for Local Area Networks", ACM Transactions on Computer Systems, pp.319-352, Nov. 1993、参照）。その考え方は、スイッチの各出力ポートごとに別々のキューを設け、空き出力ポート宛のパケットが、別のポートに対する競合により進めない先頭パケットによってサービスをブロックされる可能性がなくなるようにするというものである。この場合、Ｎ×Ｎスイッチは入力ポートごとにＮ個のキュー、すなわち、N^2個のキューを有する。他の研究者によって議論されているように（A. Mekkittikul, N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Through-put in Input-Queued Switches", Proceedings of Infocom98, April 1998、参照）、ＶＯＱ法をさらに研究することによって、高性能のスケジューラの設計により実に１００％のスループットが達成可能であることが示されている。

【０００４】従って、ＶＯＱ入力バッファ型スイッチのスケジューラは高速入力バッファ型スイッチの重要な設計ポイントのうちの１つとなる。ＶＯＱの場合、スケジューラは、通常の先入力先出力（ＦＩＦＯ）入力キューイングアーキテクチャの場合よりも、バックログのある入力ポートから出力ポートへパケットを交換するのにはるかに多くの選択肢を有する。バックログのある入力ポートのうちで、あらゆる入出力ポート対を選択することができる。

【０００５】このようなスケジューラに対する研究のほとんどは以下のように分類することができる。集中スケジューラは、スケジューラがN^2個のすべてのＶＯＱに関する情報を有する単一のエンティティであって、パケットスロットごとにすべての可能な入出力ポート対に関するスケジューリング決定を行うものである（例えば、A. Mekkittikul, N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Through-put in Input-Queued Switches", Proceedings of Infocom98,April 1998、参照）。

【０００６】他方、分散スケジューラは、スケジューラがいくつかの機能ブロック（通常は入力あるいは出力ポート当たり１又は２個のブロック、あるいは、入出力クロスポイント当たり１ブロック）に分割されたものである（例えば、N. McKeown, M.Izzard, A. Mekkittikul, W. Ellersick, M. Horowitz, "The Tiny Tera: A Packet Switch Core", IEEE Micro, Jan/Feb 1997, pp.26-32、および、Y. Tamirand H-C Chi, "Symmetric Crossbar Arbiters for VLSI Communication Switches", IEEE Transactions on Parallel and Distributed Systems, Vol.4, No.1,pp.13-27, 1993、参照）。

【０００７】図１は、集中スケジューラの一例を示す概略的ブロック図である。集中スケジューラは、スケジューリング決定を行う前に、N^2個の情報にアクセスする必要がある。このようなスケジューラは、一般に、スケジューラを実装するハードウェアがスイッチラインの数Ｎに強く依存するという意味で、スケーラブルではない。

【０００８】分散スケジューラは、ハードウェアに対してスイッチポート数への依存性をより少なくする可能性を有する。しかし、これまで提案されているものは依然として、個々のパケットスロットに対するスケジューリング決定を行うことができる前に、N^2個のすべてのキューに関する情報を提供する通信メカニズムを必要とする。この通信は、並列に（ＳＬＩＰスケジューラの場合のように。N. McKeown, M. Izzard, A. Mekkittikul, W. Ellersick, M. Horowitz, "The Tiny Tera:A Packet Switch Core", IEEE Micro, Jan/Feb 1997, pp.26-32、参照）、あるいは、ラウンドロビン方式で（Y. Tamir and H-C Chi, "Symmetric Crossbar Arbiters for VLSI Communication Switches", IEEE Transactions on Parallel and Distributed Systems, Vol.4, No.1, pp.13-27, 1993、参照）行うことが可能である。

【０００９】図２（Ａ）及び（Ｂ）は、並列方式及びラウンドロビン方式のアーキテクチャをそれぞれ示す模式図である。図２（Ａ）における並列通信アーキテクチャでは、各ブロックがスイッチのサイズに陽に依存してしまう。各ブロックがN^2個のメッセージを受け取らなければならないからである。図２（Ｂ）におけるラウンドロビンアーキテクチャはこの問題点を克服するが、別の問題点を生じる。すなわち、すべての出力ポートに関するスケジューリング決定を達成するためには、メッセージ受渡しは、単一のパケットスロット中に１ラウンドを完了しなければならないという点である。これは、スケジューリング決定よりも少なくともＮ倍高速なメッセージ処理を必要とする。

【００１０】さらに最近になって、ラウンドロビン・グリーティ・スケジューラ（ＲＲＧＳ：Round-Robin Greedy Scheduler)が提案された。これは、メッセージパッシング（受渡し）に基づくスケジューラであり、各入力ポートがスケジューリング決定を行い、この情報をラウンドロビン方式で次のポートに渡すものである（本出願人による特願平１１－１７２５８４号を参照）。メッセージ受渡し速度要求条件を緩和するために、ＲＲＧＳは

パイプライン機能を導入している。入力ポートは、十分将来のスロットに関するスケジューリング決定を行い、メッセージ受渡しメカニズムがこの情報を他の入力ポートに広めるのに十分な時間があるようにする。ＲＲＧＳは高速なスケジューリングを実現することができる。

【００１１】先ず、一般的なパイプライン型スケジューラのアーキテクチャについて説明する。図3は入力バッファスイッチアーキテクチャを例示する模式図である。スイッチアーキテクチャに関して、スケジューリングは、純粋なノンブロッキングＮ×Ｎクロスバスイッチに適用されると仮定する。また、仮想出力キュー（ＶＯＱ）を用いてＨＯＬブロッキング問題に対処すると仮定する。

【００１２】さらに、固定サイズパケットおよび一様リンク速度を仮定する。時間はスロット化される．１つのスロットは、出力リンクによる１パケットの送信にかかる時間として定義される。出力ポート競合が存在しない場合、ノンブロッキングクロスバは、タイムスロット当たりＮパケットまでを交換することができる。スケジューラの基本的な仕事は、スロットごとに、空でないN^2個のＶＯＱキューのうちのいずれが出力ポートにアクセスすることができるかを判断することである。効率のために、スケジューラは、１タイムスロット内でバックログのあるキューの間のすべての競合を解決しなければならない。

【００１３】ライン速度が増大すると、スケジューリングアルゴリズムが大容量スイッチにもスケーラブルであることが重要となる。従って、分散アーキテクチャが有力であると思われる。分散アーキテクチャでは、高速スイッチにおいてパケットスケジューリングに要求されるきつい処理時間が緩和されるからである。例えば、１０Ｇｂｉｔ／ｓのライン速度の１６×１６ポートスイッチで、スケジューリング決定は、各パケット送信時に行わなければならず、４２４ビットのＡＴＭセルに対して４２ｎｓである。シーケンシャルスケジューラを使用する場合、各決定は、１６×１６スイッチでは０．１６ｎｓ未満で行わなければならない。N^2個の決定をしなければならないからである。光コアを使用する場合、光コアの全交換帯域幅要求条件をそのままにして、電子ハードウェアをポートごとに分散することには意味がある。さらに、分散スケジューラは、当然、任意のライン数にあわせてスケールされる。図4にそのようなスケジューラを例示する。

【００１４】図4において、各クロスバ入力ポートは、入力ポートスケジューラモジュール（ＳＭ：Scheduler Module）を有する。各ＳＭは、個別のＩＤであるＳＭ－ＩＤを有する。ライン数とのスケーラビリティを維持するために、ＳＭは、隣の１個のＳＭとのみ通信することが許される。これにより、ＳＭハードウェアブロックは任意のＮ×Ｎクロスバファブリックで使用可能であることが保証される。ＳＭ通信チェインが図4に示されている。これは、タイムスロット、スロット所有権、および出力ポート予約のようなスケジューリング情報を通信するために使用される。クロスバモジュールとＳＭとの間の唯一の相互作用は、グローバルクロックを通じてのものである。これは、あらゆるＳＭに、どのスロットが現在のシステムタイムスロット（ＣＴＳ：Current system Time Slot）であるかということと、ＣＴＳで交換される入出力ポート対に関する現在の決定テーブル（図示せず）とを知らせる。これは、スケジューラによって書き込まれ、クロスバファブリックによって読み出されるグローバルメモリによって実現することが可能である。

【００１５】タイムスロットごとに、各ＳＭは、アクセス要求先の出力ポートに関して完全な選択の自由があると仮定される。同様の選択をするＳＭどうしは「コリジョン」（衝突）を生じ、これは、与えられたスロットに対するグローバルスケジューリングパターンを決定する前に解決する必要がある。ＳＭが、他のすべての要求に関する現在の情報を有することになる場合、通信チェインは、スケジューリング決定の速度よりもＮ倍速い速度で動作しなければならない。すなわち、ＳＭは、１つのスケジューリング決定を行う前に、Ｎ個のメッセージを受信することができなければならない。

【００１６】ＳＭハードウェアの速度をライン速度とともにスケーラブルに保つために、Ｎルックアヘッド（先読み）スケジューリング方式を使用することが可能である。すなわち、各ＳＭは、現在のスロットの少なくともＮスロット先のタイムスロットに関してスケジューリング決定をすることになる。この機能により、ＳＭは、スケジューリング決定をする前に、同じタイムスロットに対してなされている他のスケジューリング決定に関して知っていることが保証される。さらに、この機能は、通信チェインを入力ライン速度のＮ倍に高速化する必要がない。ＲＲＧＳは上記のような、分散スケジューリング、パイプラインスケジューリングの特徴と、Ｎルックアヘッド（先読み）スケジューリングの特徴を備えている。

【００１７】図5は、４×４クロスバースイッチを用いた場合のＲＲＧＳスケジューリングの一例を示すタイムチャートである。図５では、４個のＳＭ１～ＳＭ４と、それらの入力が出力ポートを選択するタイムスロットＴ６、Ｔ７・・・との関係が示されている。

【００１８】図5において、例えばタイムスロットＴ５で、ＳＭ１はタイムスロットＴ１０で送信を行うための出力ポートの選択（スケジューリング）を行い、ＳＭ３はタイムスロットＴ９におけるスケジューリングを行っている。また、次のタイムスロットＴ６では、ＳＭ１はタイムスロットＴ８におけるスケジューリングを行っている。以下同様である。

【００１９】上記のように各ＳＭがスケジューリングを

行い、その結果を次段のSMに転送することによって、あらゆるSMが、既にスケジューリングされたポートに関する情報を適時に得ることが保証される。あるSMが、前の「訪問者（visitor）」（即ち、1タイムスロット前のSM）によって既に選ばれた出力ポートを選ぶことを避ければ、コリジョンを完全に回避することができる。

【0020】

【発明が解決しようとする課題】しかしながら、RRGSでは、一つのSMが予約を行っていくタイムスロットの巡回順序が複雑になる。図6は、図5を個々のSMのタイムスロット巡回順序に着目して表現したタイムチャートである。例えば、SM1について見ると、タイムスロット巡回順序は、T10、T8、T11、T9・・・となり、時系列的あるいは逆時系列的な一定の規則的順序にはなっていない。これはRRGSの実装及び制御が複雑になるという問題を示している。

【0021】さらに、前記特願平11−172584号に示されるように、RRGSは、SMが偶数の場合と奇数の場合でスケジューリング動作が異なる。これは、SMを追加する際に制御を変更しなければならないことを示しており、実装及び制御が複雑になるという問題がある。

【0022】また、RRGSでは、SMが、まだ選ばれていない出力ポートを選ぶように制限されるため、VOQサービスレートは予測が困難になる。さらに、重大な公平性の問題が生じる。例えば、図4において、SM#1とSM#2は与えられた出力ポートのキューにコンスタントにバックログがあり、他のSMの対応するキューは空であるとする。この場合、SM#1は、図5にて定義される巡回順序においてSM#2の前に4スロットのうちの3スロットを訪れるため、4スロットのうち3スロットはSM#1によって取られることになる（前記特願平11−172584号を参照）。

【0023】このように、上記のRRGSスケジューラは高速なスケジューリングを実現することが可能であるが、実装及び制御が複雑になるという問題がある。また、予測可能かつ調整可能なサービスレートを実現できない。また、上述したようにVOQのいくつかが他のVOQの状態によりスケジューリングを妨げられるという公平性の問題もある。

【0024】本発明の目的は、RRGSの実装及び制御の複雑さを解消した簡易なスケジューラの基本方式を提供することにある。

【0025】本発明の他の目的は、VOQキューがスケジューリングを行う際の完全な選択の自由を可能にする大容量スイッチのためのスケジューラを提供することにある。

【0026】本発明のさらに他の目的は、VOQサービスレートを予測可能かつ調整可能にするスケジューラを提供することにある。

【0027】本発明のその他の目的は、いずれのVOQも他のVOQの状態に拘わらず同じ確率でスケジューリングされるという意味で公平であるようなスケジューラを提供することにある。

【0028】スケジューラ設計におけるもう1つの制約は、次にスケジューリングされる与えられた入線に属するN個のVOQのうちのどのVOQを、スケジューラの制御から外すかの決定である。換言すれば、入力ポートごとに、どの出力ポートが次にスケジューリングされるかを、ある外部エンティティが全く自由に決定することである。この要請は将来のサービス品質（QoS）のサポートにとって重要である。これによりVOQの予測可能なサービスレートをより予測可能にする最大スループットを低下させる可能性があることは明らかである。しかしながら、これは重要な点である。スイッチ全体のスループットの最大化は、一部のキューの枯渇、ひいては、それらのキューに関連するフローも妨げられる可能性があるからである。

【0029】

【課題を解決するための手段】本発明の第1の観点によれば、ネットワークにおけるデータのフローを制御するスイッチは、複数の入力ポートと、複数の出力ポートと、複数の入力ポートスケジュールモジュールを有するスケジューラとを有する。各スケジュールモジュールは、前記複数の出力ポートのうちの指定された出力ポートへデータを送るように、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングする。スケジュールモジュールは、モジュール間でスケジューリングメッセージを受け渡し、各スケジュールモジュールは、当該スケジュールモジュールが指定出力ポートにアクセスしようとする将来のタイムスロットを計算する。スケジュールモジュールは、更に、前記将来のタイムスロットが当該スケジュールモジュールによって現在予約されているかどうか、前記将来のタイムスロットが阻止されているかどうか、及び前記将来のタイムスロットが他のスケジュールモジュールによって取られているかどうかに基づいて、前記将来のタイムスロットが有効かどうかを判断する。有効な場合、スケジュールモジュールは前記将来のタイムスロットを取り、スケジューリングメッセージに前記将来のタイムスロットが取られたことを示す情報を入れる。

【0030】スイッチのスケジューラは、前記将来のタイムスロットが予約されているとき又は取られているときに、前記将来のタイムスロットを所定数のタイムスロットだけ前進させる。

【0031】スイッチは、前記出力ポートのそれぞれに対して別々のキューを維持する仮想出力キューイングを用いて、前記入力ポートを通じてデータ入力をキューイングする。あるいは、個々のポートに対する仮想出力キ

ューイングは、他のポートに対する仮想出力キューイングとは独立であることも可能である。さらに、スイッチは予測可能かつ調整可能な仮想出力キューイングのサービスレートを有する。また、スイッチスケジューラは重み付きラウンドロビンに基づいて指定出力ポートを選択する。

【００３２】本発明の第２の観点によれば、スイッチの複数の入力ポートに到着する入力パケットをスイッチの複数の出力ポートに送るようにスケジューリングする方法が提供される。ここで、スケジューラは、複数の入力ポートスケジュールモジュールを有する。この方法は、

ａ）現スケジュールモジュールが、前のスケジュールモジュールからスケジューリングメッセージを受信するステップと、

ｂ）前記現スケジュールモジュールが、前記複数の出力ポートのうちの１つにアクセスしようとする将来のタイムスロットを計算するステップと、

ｃ）前記将来のタイムスロットにおける送信用にスケジューリングするように前記複数の出力ポートのうちの１つを選択するステップと、

ｄ）前記将来のタイムスロットが前記現スケジュールモジュールによって既に予約されているかどうかを判断するステップと、

ｅ）前記将来のタイムスロットが前記現スケジュールモジュールによって予約されていない場合、前記将来のタイムスロットが阻止されているかどうかを判断するステップと、

ｆ）前記将来のタイムスロットが阻止されていない場合、前記将来のタイムスロットが他のスケジュールモジュールによって既に取られているかどうかを判断するステップと、

ｇ）前記将来のタイムスロットが、他のスケジュールモジュールによって既に取られているか又は前記現スケジュールモジュールによって既に予約されている場合、前記スケジューリングメッセージから、繰越し動作が既に開始されているかどうかを判断するステップと、

ｈ）前記繰越し動作が既に開始されている場合、前記将来のタイムスロットを阻止状態に設定してステップｄに戻るステップと、

ｉ）前記繰越し動作が開始されていない場合、前記将来のタイムスロットを所定数のタイムスロットだけ前進させ、繰越しフラグをセットして、ステップｄに戻るステップと、

ｊ）前記将来のタイムスロットが他のスケジュールモジュールによって取られていない場合には前記将来のタイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、

ｋ）前記スケジューリングメッセージを次のスケジュールモジュールに渡すステップと、からなる。

【００３３】複数の入力ポートを通じて入力するデータは、各出力ポートに対して別々のキューを維持する仮想出力キューイングを用いてキューイングされる。個々のポートに対する仮想出力キューイングは他のポートに対する前記仮想出力キューイングとは独立である。また、前記仮想出力キューイングのサービスレートは予測可能かつ調整可能である。スケジューラは、重み付きラウンドロビンに基づいて、前記指定出力ポートを選択する。

【００３４】本発明の第３の観点によれば、ネットワークにおけるデータのフローを制御するスイッチは、複数の入力ポートと、複数の出力ポートと、前記複数の出力ポートのうちの指定出力ポートへデータを送るように、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングするＮ個の複数の入力ポートスケジューリングモジュールを有するスケジューラと、から構成され、前記スケジューラは、前記各入力ポートスケジューリングモジュールがリング状に接続され、タイムスロット単位に、各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから、ある予約タイムスロットの予約状況情報を受信し、各入力ポートスケジューリングモジュールが、その予約タイムスロットにおける当該入力ポートスケジューリングモジュールからのパケット送出予約可否を決定し、各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから受信した予約状況情報に、自スケジューリングモジュールの予約結果を反映させて、次段のスケジューリングモジュールに送信する。

【００３５】複数の入力スケジューリングモジュールを有するパケットスイッチの入力ポートと出力ポートの接続状態を決定し接続を予約する（以下、スケジューリングという。）方法は、Ｎ個のタイムスロットを単位とするフレームを定義して、前記フレーム時間内で、前記フレームの次フレーム中のＮ個のタイムスロットでのスケジューリングを行うことを特徴とする。

【００３６】上記スケジューリング方法は、

ａ）現在のスケジュールモジュールが、前のスケジュールモジュールからスケジューリングメッセージを受信するステップと、

ｂ）前記現在のスケジュールモジュールが、前記複数の出力ポートのうちの１つにアクセスしようとする将来のタイムスロットを予め次フレーム内の特定のタイムスロットに決定するステップと、

ｃ）前記複数の出力ポートのうちの１つを前記将来のタイムスロットでの送信用にスケジューリングするように選択するステップと、

ｄ）前記将来のタイムスロットが他のスケジュールモジュールによって既に取られているかどうかを判断するステップと、

ｅ）前記将来のタイムスロットが他のスケジュールモジュールによって取られていない場合には、前記将来のタ

イムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、
ｆ）前記スケジューリングメッセージを次のスケジュールモジュールに渡すステップと、からなる。
【００３７】さらに、上記スケジューリング方法は、タイムスロットでのスケジューリング決定過程（接続決定過程）の観点から見た場合、前記複数の接続決定過程が、フレームの先頭で同時に開始され、フレーム内で同時にパイプライン処理により進行し、フレームの末端で同時に完了する。
【００３８】また、このスケジューリング方法は、前記入力ポートスケジューリングモジュールが、フレームの先頭で同時に開始する前記各接続決定過程において、次のフレーム内の各々異なる予約タイムスロットを対象として処理を開始する。
【００３９】上記スケジューリング方法は、ある与えられた将来のタイムスロットにおけるＮ個の入力ポートスケジューリングモジュールの規則的な巡回順序を定義するＮ×Ｎマトリクスを参照することによって、現在のフレームにおける入力信号が次のフレームでどの出力ポートへ送出されるかを決定する。
【００４０】
【発明の実施の形態】本発明によるキャリーオーバ（繰越し）ラウンドロビン・パイプライン・スケジューラ（Carry Over Round-robin Pipelined Scheduler、以下ＣＯＲＰＳという。）は、高速クロスバファブリックに対する公平なスケジューラであり、従来技術のスケジューラの問題点を解決するものである。ＣＯＲＰＳは、高速スイッチファブリックのライン速度およびライン数の双方に関するスケーラビリティ性を有する。ライン数に関するスケーラビリティのために、メッセージ受渡しを有する分散アーキテクチャが選択される。更に、ＲＲＧＳと同様に、メッセージ処理要求条件をライン速度とともにスケーラブルに維持するために、パイプラインアーキテクチャが用いられる。
【００４１】さらに、本発明によれば、時間軸を単にＮ個の連続するスロット列であるスロットフレームに分割して、時間をフレームの列とみなす。競合するスケジューラモジュールＳＭ間のコリジョンを解決する基準を設定するために、優先マトリクスを使用する。Ｎ×Ｎ優先マトリクスは、将来の与えられたタイムスロットをＳＭが巡回する順序を定義するマトリクスである。マトリクスの行は現在のフレーム（現在のシステムスロットを含むフレーム）内のスロットをインデクス付けし、マトリクスの列は次に訪れるフレーム内のスロットをインデクス付けしている。マトリクスの要素は、どのＳＭが、列インデクスによって示される次フレーム内のスロットを「訪れる(visit)」べきかを指定する。
【００４２】図７は４×４優先マトリクスを示す図であり、図８は、図７に示したマトリクスに対するパイプライン化されたタイムスロット巡回順序を例示している。パイプライン化された決定プロセスは、優先マトリクスの使用に既に含まれていることに注意すべきである。例えば、システムの現在のタイムスロットが現フレームの第２スロットであるとき、ＳＭ＃１が次フレームの第４スロットに関するスケジューリング決定をしている間に、ＳＭ＃３は次フレームの第２スロットに関するスケジューリング決定をしている。
【００４３】優先マトリクスを使用することにより時間軸がフレーム化され、タイムスロット巡回順序が規則的になる。例えば、フレームＦ１における各ＳＭの動作に着目すると、各ＳＭのスケジューリング決定過程は、フレームＦ１の先頭で同時に開始され、スケジューリング決定を行うタイムスロットの巡回順序はＴ８→Ｔ７→Ｔ６→Ｔ５→Ｔ８となり、フレームＦ１の末端で同時に完了する。これは、図６に示したＲＲＧＳにおけるタイムスロット巡回順序（Ｔ１０、Ｔ８、Ｔ１１、Ｔ９・・・）と比較して規則的になっている。このためＳＭの実装及び制御が容易になる。更に、Ｎ×Ｎ優先マトリクスはＳＭの個数に関して偶奇の区別なく、同一規則にてタイムスロットの巡回順序を定義する。
【００４４】図９は５×５優先マトリクスを示す図であり、図１０は、図９に示したマトリクスに対するパイプライン化されたタイムスロット巡回順序を例示している。フレームサイズは５タイムスロットとなるが、Ｎ＝４の場合と同様に、タイムスロット巡回順序は規則的なものとなる。図８と同様に、フレームＦ１における各ＳＭの動作に着目すると、各ＳＭのスケジューリング決定過程は、フレームＦ１の先頭で同時に開始され、スケジューリング決定を行うタイムスロットの巡回順序がＴ１０→Ｔ９→Ｔ８→Ｔ７→Ｔ６→Ｔ１０となり、フレームＦ１の末端で同時に完了する。ＳＭの個数に関して偶奇の区別なく同一規則にてＮ×Ｎ優先マトリクスを規定できるため、ＲＲＧＳと比較してＳＭの実装及び制御が容易になる。
【００４５】Ｎ×Ｎ優先マトリクスは、通信チェインメッセージ受渡しの同じ方向に、ＳＭの列を回転させることによって生成される。これにより、あらゆるＳＭが、既にスケジューリングされたポートに関する情報を適時に得ることが保証される。あるＳＭが、前の「訪問者(visitor)」（即ち、１タイムスロット前のＳＭ）によって既に選ばれた出力ポートを選ぶことを避ければ、コリジョンを完全に回避することができる。
【００４６】以上のように、時間軸をフレーム化して優先マトリクスを用いた予約を行うことにより、ＳＭの個数に関して偶奇の区別なくタイムスロット巡回順序が規則的なものとなり、ＳＭの実装及び制御が容易になる。
【００４７】更に、本発明では、分散アーキテクチャ及びメッセージ受渡し方式を維持しつつ公平性を提供する

ために、繰越し（キャリーオーバ）動作を導入する。この考え方は、あるスケジューラモジュールＳＭａが、それに先行するスケジューラモジュールＳＭによって所望の出力ポートが既に予約されているスロットを訪れるときに、処理しようとしたスロットから将来にＮスロット分だけそのポートのスケジューリング試行を繰り越すというものである。もし当該スロットが同じ出力ポートに取られていることが分かれば、ＳＭは所望の出力ポートがまだ取られていないスロットを見つけるまで更にＮスロット先に進む。

【００４８】図１１は、複数のスケジューラモジュールＳＭの間での繰越し動作を例示する説明図である。繰越し動作は、与えられたタイムスロットにおいて「衝突している」ＳＭの個数に依存してＮフレームまで広げることができる。繰越し動作によって影響されるスロットは、コリジョン（衝突）を解決するために取られるスロットの集合（以下、コリジョン解決セットという。）とみなすことができる。なお、繰越し動作を受けたスロットは、後続するフレームですべてのＳＭにより再び訪問されるであろう。従って、繰越し動作によって取られるスロットは、潜在的に新たなコリジョンを受け、コリジョン解決セットの重畳を引き起こす可能性がある。これは、複数のコリジョンを解決するためにN^2個のフレーム、すなわち全部でN^3個のスロットを必要とする可能性がある。

【００４９】システムのメモリ要求条件を緩和すると共にスケジューリング遅延を短縮するために、繰越し動作によって影響されるフレームの個数は、繰越し動作を実行したＳＭが、当該コリジョンを解決するために取られた複数のスロットにわたって同じ出力ポートに対して他のスケジューリングをしないように制限される。換言すれば、１つのスロットが、同時に複数のコリジョンを解決するようには使用されない。

【００５０】例えば、ＳＭａが、与えられたポートｐによって取られたスロットｍを見つけ、これにより繰越し動作がトリガされると仮定する。この繰越し動作の結果としてＳＭａによって予約されたスロットをｍｘとする。スロットｍｎ（１≦ｎ＜ｘ）のいずれも、同じポートｐについてはＳＭａにとって利用不可（阻止、ブロッキング）となる。従って、この阻止機能は、与えられたスロットに関する複数のコリジョンが禁止されることを保証する。

【００５１】ＣＯＲＰＳスケジューリングアルゴリズムについて以下で説明する。先ず、通信チェインで渡されるメッセージと、スケジューリング決定が記録されるＳＭデータベースについて説明し、その後、アルゴリズムの流れについて説明する。

【００５２】各セルスロットで、チェイン内のあるＳＭから次のＳＭに渡されるスケジューリング決定要素のベクトルを定義する。Ｓメッセージは、たかだか最後のＮ個のセルスロットでなされたスケジューリング決定のスケジューリング要素（scheduling element、以下ＳＥと記す。）を含む。すなわち、Ｓメッセージは、たかだかＮ個のＳＥを有する。Ｓメッセージは以下のフォーマットを有する。

【００５３】図１２は、Ｓメッセージのフォーマットを示す図である。同図において、Ｓメッセージの各スケジューリング要素ＳＥは、存続時間(ＴＴＬ：Time To Live)、タイムスロットＩＤ（ＴＳＩ：Time Slot ID）、ＳＭ－ＩＤ、及び出力ポートＩＤ（ＯＰＩ：Output Port ID）からなる。

・存続時間(ＴＴＬ)は、当該ＳＥを生成したＳＭによって最初にＮにセットされる。

・タイムスロットＩＤ（ＴＳＩ）は、現在のＴＳ（タイムスロット）からスロットがスケジューリングされるまでのスロット数として定義される、スケジューリングされるスロットのＩＤである。

・ＳＭ－ＩＤは、スケジューリング予約をした入力ポートスケジューリングモジュールのＩＤである。

・出力ポートＩＤ（ＯＰＩ）は、スケジューリングされる出力ポートのＩＤである。

【００５４】スロットの最初に、各ＳＭは先行するＳＭからＳメッセージを受信する。これは、最後のＮ個のスロットに付けられたＳＥを含む。あらゆるＳＭは、タイムスロット当たりたかだか１回のスケジューリング決定を行う。ＳＭｐがスケジューリング決定を行う場合、ＳＭｐは以下の内容を有する新しいＳＥを作成する。

【００５５】・ＴＴＬ＝Ｎ

・ＴＳＩ＝現在のスロットから選択されたタイムスロットまでの（それを含む）スロットの個数ｍ

・ＳＭ－ＩＤ＝ｐ

・ＯＰＩ＝タイムスロット（ＣＴＳ＋ｍ）におけるパケットが入力ポートｐから出力ポートｑに交換されるような所望の出力ポートｑ。

【００５６】スケジューリング決定にかかわらず、各ＳＭは、次のＳＭにメッセージを渡す前に、Ｓメッセージ内の他のすべてのＳＥのＴＴＬをデクリメントし、ＴＴＬ＝０のＳＥを廃棄する。

【００５７】各ＳＭは、（Ｎ＋１）Ｎ個の位置を含むメモリアレイＳＣを有する。最初のＮ個の位置は、クロスバスイッチモジュールによって読み出される現フレームのスケジューリング決定を記録する。これらの位置は、すべてのＳＭの間で、現フレームに関する同一の情報を有し、いくつかの方法でクロスバコントローラによってアクセスされ得る。厳密にいえば、ＳＭは、この情報を保持する必要はない。残りのN^2個の位置は、将来のスケジューリング決定を記録するために使用される。メモリアレイは以下のフォーマットを有する。

【００５８】図１３に示すように、以下のフィールドが定義される。

・タイムスロットＩＤ：ＳＣアレイへのインデクスである。これは、ＳＣ位置がスケジューリング情報を保持するタイムスロットＩＤを与える。これは、クロスバモジュールによって提供されるグローバルクロックと同期する。このフィールドは、グローバルクロックが進行すると共にラップアラウンドする。

【００５９】・阻止（Blockage）：これはＳＭがスケジューリング予約をすることを阻止されている出力ポートの集合を定義する。このフィールドにはＮ個までのエントリが存在しうる。なお、最初は空である。

【００６０】・予約（Reservations）：これは与えられたタイムスロットに対するスケジューリング予約を記録する。ＣＯＲＰＳは、現タイムスロット（ＣＴＳ）に対してこのフィールド内のすべてのエントリがすべてのＳＭにわたり同一であることを保証する。従って、クロスバモジュールは、任意のＳＭ（ＣＴＳ）から、セルの現在の入出力スケジューリングを読み出すことができる。アルゴリズムの一貫性チェックは、クロスバコントローラが十分な処理時間を有する場合に、すべてのＳＭの間でこのフィールドを比較することにより、クロスバモジュールによって実行されることができる。

【００６１】ＣＯＲＰＳスケジューリングアルゴリズム

各ＳＭは、ここで説明するＣＯＲＰＳスケジューリングアルゴリズムに従う。ＣＯＲＰＳには、与えられたＳＭがどの出力ポートをスケジューリングしようとするかに関する制約はない。どの出力ポートをスケジューリングしようとするかの選択は、各ＳＭにまかされ、そのＶＯＱにサービスする固有のポリシーに従う。図１４はＣＯＲＰＳスケジューリングアルゴリズムを示すフローチャートである。以下、タスクボックス１０１～１１０について、同図のフローに従いながら説明する。

【００６２】先ず、タスク１０１において、前のＳＭからＳメッセージを受信し、各ＳＥに対してＴＴＬ（存続時間）をデクリメントする。更に、与えられたＳＥに対してＴＴＬ＞０の場合には、ＴＳＩ（タイムスロットＩＤ）をデクリメントし、ＴＳＩにおけるメモリアレイＳＣを更新する。ＴＴＬ＝０の場合には、そのＳＥをＳメッセージから除去する。また、キャリー（ＣＡＲＲＹ）フラグをＣＡＲＲＹ＝ＦＡＬＳＥにリセットする（タスク１０９参照）。

【００６３】続いて、タスク１０２（試行スロットの計算）において、適当な優先マトリクスを用いて、どの将来のタイムスロット（ＦＴＳ：Future Time Slot）をスケジューリングしようとするかを計算する。簡単のため、そのマトリクスは、ＦＴＳ＝ｆ（ＣＴＳ，ＳＭ＿ＩＤ）の形の関数ｆにエンコードされうる。

【００６４】更に、ＳＭが送信用にどの出力ポート（ＯＰＩＳ）をスケジューリングしようとするかを選択する（タスク１０３：出力ポートの選択）。なお、出力ポートを選択するストラテジは、前のタスクの結果に依存する可能性がある。ＣＯＲＰＳはこのストラテジを指定しない（例えば、出力ポートの重み付きラウンドロビン選択が使用可能である）。

【００６５】続いて、ＳＣ（ＦＴＳ）の予約エントリのうちで、ＳＭ－ＩＤがこのスケジューリングを実行するＳＭに等しいものがあるかどうかを単にチェックする（タスク１０４：スロットを自分が所有しているかのテスト）。

【００６６】もし、ＳＭ－ＩＤがこのスケジューリングを実行するＳＭと異なるならば（タスク１０４のＮＯ）、更に、ＳＣ（ＦＴＳ）の阻止エントリのうちで、ＯＰＩ（出力ポートＩＤ）が、スケジューリングを試みている出力ポートＯＰＩＳに等しいものがあるかどうかをチェックする（タスク１０５：自分は阻止されているかのテスト）。

【００６７】自分が阻止されているならば（タスク１０５のＹＥＳ）、Ｓメッセージを次のＳＭに渡す（タスク１０６）。

【００６８】自分が阻止されていないならば（タスク１０５のＮＯ）、ＳＣ（ＦＴＳ）の予約エントリのうちで、ＯＰＩ（出力ポートＩＤ）が、スケジューリングを試みている出力ポートＯＰＩＳに等しいものがあるかどうかをチェックする（タスク１０７：そのスロットは取られているかのテスト）。

【００６９】そのスロットが取られていない場合には（タスク１０７のＮＯ）、ＳＣ（ＦＴＳ）に、自己のＳＭ－ＩＤを有しＯＰＩがＯＰＩＳに等しい予約エントリを作成し、ＴＴＬ＝Ｎ、ＴＳＩ＝ＦＴＳで、ＳＭ－ＩＤは自己のＩＤに等しく、ＯＰＩ＝ＯＰＩＳであるＳＥを作成する（タスク１０８：スロットを取る）。その後、タスク１０６（Ｓメッセージの受け渡し）へ進む。

【００７０】このスロットを自分が所有している場合（タスク１０４のＹＥＳ）あるいはそのスロットが既に取られている場合（タスク１０７のＹＥＳ）には、繰越し動作が既に開始されているかどうかを検査し、フラグＣＡＲＲＹ＝ＴＲＵＥ／ＦＡＬＳＥをチェックする。ＣＡＲＲＹ＝ＴＲＵＥの場合には、ＳＣ（ＦＴＳ）の阻止フィールドに、ＯＰＩ＝ＯＰＩＳのエントリを作成し、そうでない場合にはＣＡＲＲＹ＝ＴＲＵＥにセットし、更に、ＦＴＳ＝ＦＴＳ＋Ｎにセットする（タスク１０９：繰越し）。

【００７１】タスク１０９により繰越し動作が実行されると、続いて健全性チェックが行われる（タスク１１０）。即ち、ＦＴＳは、ＣＴＳから$2N^2$より遠くに離れてはならない。（ＦＴＳ－ＣＴＳ）＞$2N^2$である場合（タスク１１０のＮＯＫ）、エラーメッセージを出して処理を中止する。ＦＴＳ－ＣＴＳ）≦$2N^2$の場合には（タスク１１０のＯＫ）、タスク１０４へ戻る。

【００７２】ＣＯＲＰＳアルゴリズムを用いることで以下の利点が生じる。バックログのあるＶＯＱは、最終的

にはそのSMによって選択されると仮定しても、枯渇することはない。VOQqがSMpによって選択されると仮定すると、図14によれば、SMpがqをスケジューリングすることに成功せずに予約ループを抜ける唯一の場合は、試みたスロットに対して阻止されている場合である。SMpが阻止されているとは、キューqが既にスケジューリングされていることを意味するが、以下の点に注意すべきである。ループを抜ける他の唯一の場合があるとすれば、健全性チェックを通る場合であるが、これは、繰越し動作が次のNフレームに空きスロットを見つけなかったことを意味する。1つのコリジョンにかかわるSMはたかだかN個であり、複数のコリジョンは阻止手続きによって禁止されるため、ループからこのようにして抜けることはない。

【0073】同じ出力ポートqを連続してスケジューリングしようとするm個の入力ポート（SM）のセットをMとする。さらに、Δtのタイムスロットの間に出力ポートqに対してSMiによってスケジューリングされるスロットの個数をn_i^q（Δt）とする。スケジューラは、任意の期間Δt及びi，j∈Mに対して、

$|n_i^q(\Delta t) - n_j^q(\Delta t)| \leq N$

である場合、m－フェア（m－公平）であるという。換言すれば、SMは、他のいずれのSMよりも前にN個の予約をすることはできない。

【0074】CORPSは、m－公平（1＜m≦N）である。与えられたスロットtsにおいて出力ポートqに対してm個のSMが衝突していると仮定する。衝突しているm個のSMはそれぞれ、そのスロットに対して阻止されてはいない。もし阻止されていれば、そのスロットが既に取られているかどうかのテストをすることさえできないからである（図14のタスク105）。これらのm個のSMがスロットtsにおいて阻止されていない場合、ts＋nN（1＜n≦i）の間にm個の空きスロットがなければならない。なぜならば、将来にこれらのスロットにアクセスする唯一の方法は繰越し動作を通してであり、しかも、これらのSMはこれらのスロットに対して繰越し動作を実行していないことが分かっている（さもなければそれらはtsにおいて阻止されている）からである。このことは、次のiフレーム以内に、衝突中のSMがそれぞれqに対してスケジューリング要求をすることになることを意味する。ここで、それらは現フレームのN個の連続するスロットに対して衝突し続け、しかも、各コリジョン（衝突）はSMごとに次のiフレームに1つのスケジューリングを生成するとすれば、各SMは、出力ポートqに対して次のiフレームに全部でNスロット予約することになる。このように、コリジョンが解決されるiフレームのiN個のスロットから取られるスロットのいずれのサブセットも、他のSMよりもNスロットより多い利益を有するSMを含むことはできない。

【0075】最後の注意は興味深いものである。それは、たとえ測定期間がどれほど長くても、連続してバックログのあるVOQは、他のSMの対応するVOQの前にN個より多くのパケットをサービスされることはないことを意味するからである。実際、十分長い期間では、すべての衝突するSMは厳密に同数の予約を得ることになる。

【0076】さらに、重い負荷のもとでは、共通の出力ポートを有するキューはすべて、それらのSMによって同じ回数だけ選択されると仮定すれば、すべて同じスループットを有する（図14のタスク108）。

【0077】CORPSアーキテクチャについていくつかのコメントをしておかなければならない。SMどうしの間でスケジューリング情報を渡すために用いられる通信チェインは、将来の少なくともN個のスロットである限り、任意の方法でスロットのスケジューリングパターンを変更するために使用することが可能である。例えば、出力ポート予約を取り下げることも可能である。この機能は、SMがコリジョンにより遠い将来に予約をしたばかりであるが、ちょうど次のスロットにおいて、要求するポートが空いたことに気がついた場合に有用となる。SMが、同じパケットに対して別の予約（より近いもの）をする場合、遠いほうの予約は、取下げがなければ帯域の浪費を引き起こす。しかし、予約取下げは、上記の性質に悪影響を及ぼす可能性もある。例えば、衝突したSMが後で予約を取り下げた場合、同じコリジョンにおいてその後にスケジューリングされたパケットの遅延に悪影響を及ぼす。換言すれば、i－1個の他のSMと衝突したSMが、この衝突による予約を後で取り下げた場合、システムは、最初にi－1個のSMのみが衝突したのと同じ状態にはない。このスケジューラは、最初の設計目標を満たしながら、できる限り単純なものである。これにより、最終的な実装に要求されるハードウェアは単純なままであることが保証される。

【0078】CORPSは、複数のフレームにわたりパケットスケジューリングを広げることによって衝突を解決する。従って、他のスケジューラに比べて、平均パケット遅延が大きくなると期待することはもっともである。このため、一様トラフィックのもとでCORPSのパフォーマンスを分析する。最終目標は、繰越し動作がどのくらいパケット遅延に影響するかを評価し、競合するスケジューリングアルゴリズムと比べて、システムから最大利用率を得ることである。

【0079】トラフィック負荷に対するパケット遅延に関してスケジューラのパフォーマンスを評価するために、CORPSの分析モデルを作成する。以下では、簡単のため、次の2つの主要な仮定をする。

【0080】（i）一様トラフィック到着過程、および（ii）各SMによるランダムなVOQキュー選択（図14のタスク103）。

【００８１】与えられたＳＭｍの、出力ポートｎ宛のターゲットＶＯＱキューQ_{mn}を定義する。パケットは、強度ｐでベルヌーイ過程に従ってあらゆる入力ポートに到着する。具体的には、与えられたスロットにおいて、１つのパケットが１つの入力ポートに到着する確率がｐである。さらに、あらゆるパケットは、いずれの出力ポート宛の確率も等しい（仮定ｉ）。従って、ターゲットＶＯＱキューにおけるパケット到着過程は、パラメータｐ／Ｎのベルヌーイ分布を有する。

【００８２】ＶＯＱ選択に関して、与えられたＳＭの空でない各キューは、スケジューリングのために等確率で選択される（仮定ｉｉ）。従って、任意のＶＯＱに対して、当該ＶＯＱが空でなければ、ｑは選択される確率である。以下、Chipalkatti等（“Protocols for Optical Star-Coupler Network using WDM,” IEEE Journalon Selected Areas in Communications, Vol. 11, NO. 4, May 1993）に従うと、すべてのＶＯＱの利用率がρである場合、１つのＳＭにおいて期待される空でないＶＯＱキューの個数は$1+(N-1)\rho$によって与えられる。

【００８３】さらに、ｑと密接に関連する別の確率を導入すると便利である。ｒを任意のキューがそのスケジューラによって選択される確率とする。ｑが当該キューが空でないと仮定しているのに対して、ｒにはこの制限がないという点で、ｒはｑとは異なる。次式が成り立つのを見るのは困難ではない。

【００８４】$r=\rho q=p/N$　　（１）

【００８５】Q_{mn}のふるまいは以下のようにモデル化することができる。パケット到着間時間は明らかに、パラメータｐ／Ｎの幾何分布に従う。先頭パケットは、ＳＭによって選択されるまで待機しなければならない。その選択は、与えられたスロットにおいて確率ｑで起こる。選択された後、図１４のタスク１０５に従って、スケジューリングから阻止される可能性がある。与えられたスロットにおいてポートｍに対して阻止される確率がP_b^mである場合、先頭パケットがＳＭによって選択されるまでの待機時間は、パラメータ$s=q(1-P_b)$の幾何分布に従う。ここで確率はすべての出力ポートに対して同一であるので、上付き添字ｍを落とすことができる。Q_{mn}が選択された後、常に予約が将来のタイムスロットにおいて行われ、且つパケットはキューから一種のベルトコンベヤへと送出されると仮定する。ここでパケットは、予約タイムスロットがやって来るのを待機し、やって来た時点でシステムから出る。

【００８６】図１５は、Q_{mn}キューイングシステムに用いられるモデル全体を示す模式図である。到着パケットはまずＧｅｏ（ｐ／Ｎ）／Ｇｅｏ（ｓ）／１キューに加わる。パケットは、このキューを出ると、追加遅延D_{corps}を受ける。これは、ＣＯＲＰＳがコリジョンを解決する特定の方法の結果生じる遅延である。これは、無限個のサーバを有するボックスによってモデル化される。

【００８７】ＣＯＲＰＳを通過するパケットの期待遅延は、Ｇｅｏ（ｐ／Ｎ）／Ｇｅｏ（ｓ）／１に対する期待遅延と、平均遅延$\langle D_{corps}\rangle$との和によって与えられる（M. J. Karol, M. G. Hluchyj, S. P. Morgan, "Input Versus Output Queuing ona Space-Division Packet Switch", IEEE Transactions on Communications, Vol.COM-35, No.12, pp.1347-1356, Dec. 1987、参照）。これは、次のように書くことができる（なお、数式中上付きバーで表記している平均値は、明細書本文中で〈〉で囲んで表記しているものと同一である）。

【００８８】

【数１】

$$D=\frac{p\overline{S(S-1)}}{2N\left(1-\frac{pS}{N}\right)}+S+\overline{D_{corps}} \qquad (2)$$

ただし、Ｓは、Ｇｅｏ（ｓ）時間分布の確率変数である。次に、$\langle D_{corps}\rangle$の計算について説明する。

【００８９】Q_{mn}が選択された（先頭パケットがＧｅｏ（ｐ／Ｎ）／Ｇｅｏ（ｓ）／１を出た）後、いくつかの事象が起こり得る。まず、ＳＭｍは、試行しているスロットを所有していないことを確認しなければならない（図１４のタスク１０４）。スロットが、ＳＭによって、出力ポートｎに対して所有される確率をP_0^nとする。さらに、与えられたＳＭが、与えられたタイムスロットにおいて、出力ポートｎに対して阻止される確率をP_b^nとする。これから、次式を導くことができる。

【００９０】

【数２】

$$P_0^n=1+\frac{1}{N}\left[1+(N+1)(1-r)^N-\frac{2[1-(1-r)^{N+1}]}{r}\right] \qquad (3)$$

【００９１】ＣＯＲＰＳによれば、ＳＭが訪れているスロットは、このスロットが同じ出力ポートに対して前のコリジョンを解決するために使用されている場合に限り、そのＳＭが予約しようとするのを阻止することができる。

【００９２】その結果、ＳＭがいずれかのポートを所有する確率は、

$P_0=1-(1-P_0^n)^N$　　（５）

となる。

【００９３】図１４のタスク１０４によって生じる期待遅延$\langle D_0\rangle$は次式によって与えられる。

【００９４】

【数３】

$$\overline{D_0} = \sum_{k=1}^{N} NkP_0^k(1-P_0) = N\left[\frac{P_0 - P_0^{N+2}}{1-P_0} - (N+1)P_0^{N+1}\right] \quad (6)$$

【００９５】ＳＭｍが最初に訪れたスロットが空いている場合（図１４のタスク１０４、１０５、および１０７のテストがすべてＮＯ）、優先マトリクス方式が使用されていることにより、パケットの平均遅延〈D_{corps}〉がＮになることを見るのは容易である。〈D_{corps}〉＞Ｎで、コリジョンがない場合、少なくとも１つの予約が、将来の第２のフレームへとこぼれる。ここで、コリジョンによって受ける遅延D_cについて調べる。特定のスロットに対してｉ−１個の他のＳＭとのコリジョンが起こる場合、そのスロットに関してＳＭｍが有する優先順位に依存して、遅延D_cはＮからｉＮまでの間で変わりうる。そこで、ＳＭｍがそのスロットを訪れるｉ番目のＳＭである場合にパケット遅延がｊＮである確率をＰ［D_{corps}＝ｊＮ｜ｖ＝ｉ］とする。例えば、ｍがそのスロットを訪れる最初のＳＭである場合、

【００９６】

【数４】

$$P[D_{corps} = jN | v = 1] = \begin{cases} 1 & j = i \text{の場合} \\ 0 & \text{その他} \end{cases} \quad (7)$$

となる。

【００９７】上記の式は単に、ＳＭｍがスロットを訪れる最初のＳＭである場合、そのパケットはＮスロット遅延されるということである。コリジョンが起こらない場合、ＣＯＲＰＳスケジューラは将来の１フレームをスケジューリングするからである。次に、ＳＭｓ（ｓ≠ｍ）の任意の出力ポート（特に出力ポートｎ）に対するＶＯＱキューが空でなく、かつ、ｓによって選択される確率はｒであることを想起すると、Ｐ［D_{corps}＝ｊＮ｜ｖ＝ｉ］に対する一般式が次のようになることを見るのは困難ではない。

【００９８】

【数５】

$$P[D_{corps} = jN | v = i] = \begin{cases} 0 & j > i \text{の場合} \\ \binom{i-1}{j-1} r^{j-1}(1-r)^{i-j} & \text{その他} \end{cases} \quad (8)$$

【００９９】式（８）の上段は、ｍがスロットを訪れるｉ番目のＳＭである場合、その遅延はたかだかｉＮであるということである。下段の二項係数は、ｉ−１個のＳＭがｍの前にスロットを訪れた場合、これらのうちのｊ−１個のＳＭがｍと衝突する可能性があるということである。（D_{corps}＝ｊＮかつｖ＝ｉ）の形の事象の同時分布は、上記の表式に１／Ｎを乗じることによって容易に導出することができる。なぜなら、ＳＭｍは、スロット１≦ｉ≦Ｎのｉ番目の訪問者であることが等しく確からしいからである（図７参照）。

【０１００】次に、１つのパケットの期待遅延D_cは次のように導出することができる。

【０１０１】

【数６】

$$\overline{D_c} = \frac{N(N-1)}{2} r + N \quad (9)$$

【０１０２】ＣＯＲＰＳスケジューラにより生じる全遅延は次のようになる。

【０１０３】

【数７】

$$\overline{D_{corps}} = \overline{D_0} + \overline{D_c} \quad (10)$$

【０１０４】計算すべき最後の確率はP_bである。これは、与えられた出力ポートｎに対して、与えられたスロットにおいて、１つのＳＭが阻止される確率である。次式を示すことができる。

【０１０５】

【数８】

$$P_b^n = 1 + \frac{1}{N}\left[1 + (N+1)(1-r)^N - \frac{2[1-(1-r)^{N+1}]}{r} + 2 + r - \frac{(1-r^{N+1})}{1-r}\right] \quad (11)$$

【０１０６】Ｇｅｏ（ｓ）に対して、〈Ｓ〉＝１／ｓで、〈Ｓ（Ｓ−１）〉＝２（１−ｓ）／s^2であることに注意すると、１つのパケットがシステムで受ける全平均遅延は次のようになる。

【０１０７】

【数９】

$$D = \frac{p(1-s)}{Ns^2\left(1-\frac{p}{Ns}\right)} + \frac{1}{s} + N\left[\frac{P_0 - P_0^{N+2}}{1-P_0} - (N+1)P_0^{N+1}\right] + \frac{N(N-1)}{2}r + N \quad (12)$$

ただし、s＝q（1－P_b）である。最初の3項は、スケジューリングが行われる前の、VOQキューにおける遅延に対応する。第3項は、CORPSのパイプラインおよびコリジョン解決機能により、パケットが待機するのに必要な追加時間に対応する。

【0108】図16に、CORPSの遅延対スループットの解析的結果を、CORPSスケジューラを備えた16×16スイッチのシミュレーションと比較したものを示す。この図において、パイプラインおよびコリジョン解決方式が使用されることにより、パケットがSMスケジューラによって選択されるまでに受ける平均キューイング遅延と、CORPS遅延との間に違いがある。図から分かるように、解析的予測は、シミュレートされたシステムのふるまいと良く一致する。

【0109】この図は、負荷のすべての範囲を通じて、スケジューリング遅延がキューイング遅延よりも優勢であることを示している。非常に高い負荷の場合（キューが形成され始めるとき）にのみ、キューイング遅延が重要になる。これは、パケットがVOQキューに到着するとすぐに将来のパケットをスケジューリングすることにおいて、CORPSがうまくはたらいていることを意味する。他方、CORPSによって生じる平均遅延は、軽負荷の場合のおよそ1フレームから、負荷が0．85に達するときの約5フレームまで、増大する。

【0110】完全を期するため、図17に、16×16CORPSスイッチにおける全遅延の相補分布を示す。曲線は、シミュレーションによって得られた、負荷が0．8および0．85の場合のものである。まず、いずれのパケットも、システムを通過するのにN^2スロットより多くはかからないことに注目される。これは、CORPSでは多重コリジョンが起こることを許していないことによる。実際、分布のテールは、$N^2/2=128$付近のあたりで終わっているように見える。しかし、システムが非常に大きい負荷によって駆動される場合、パケット遅延はN^2に近づくようである。

【0111】図18は、CORPSを実現するシステムブロック図である。VOQMモジュールは、パケットを仮想出力キューVOQに入れる。また、このモジュールは、与えられたキューに代わって、SMモジュールに対して要求を行う。SMモジュールは、メッセージ受渡しを制御し、CORPSスケジューラを実現する。SMモジュールは、VOQMと通信して、将来のスロット予約について通知する。この通知はVOQMに保持され、与えられたスロットにおいて、パケットが、交換されるべきクロスバレジスタに転送されるようにする。

【0112】図中、SMとクロスバコントローラの間の通信はバスを通じて行われるように示されているが、この特定の種類の通信である必要はない。

【0113】スケジューリングアルゴリズムどうしの公平な比較では、平均遅延やスループットのようなパフォーマンス尺度のみならず、複雑さおよび実装コストも考慮すべきである。第1の選択基準は高いスループットである。さらに、VOQで動作するスケジューラのみを比較する。そこで、本発明と競合するスケジューラとして、1－SLIP及びRRGSとの比較を行う。複数回のイテレーションではなく1イテレーションのSLIPを選択する理由は、比較プロセスの公平性のためである。すなわち、任意の入力ポートにおいて、スロット当たりたかだか1回の決定をすることができると仮定する。i－SLIP（i＞1）は、実質的に、スロット当たり複数回のスケジューリング決定を要求することになる。

【0114】パフォーマンス比較において、解析的結果およびシミュレーション結果の両方をもとにする。一様トラフィックに対するRRGSおよびSLIPの遅延パフォーマンスは次のように近似することができる。

【0115】

【数10】

$$\overline{D_{RRGS}} = \frac{p(1-q)}{Nq^2\left(1-\frac{p}{Nq}\right)} + \frac{1}{q} + \frac{N}{2} \quad (13)$$

$$\overline{D_{SLIP}} = \frac{pN}{2(1-p)} \quad (14)$$

【0116】RRGSの結果については、本出願人による特願平11－172584号に記載されており、SLIPの結果については、N. McKeown, "Scheduling Cells inan Input-Queued Switch", PhD Thesis, University of California at Berkeley, 1995、に記載されている。

【0117】図19に、これらのアルゴリズムの平均遅延対スループットのパフォーマンスを、CORPSと対

照して示す。この図から明らかなように、ＲＲＧＳ及びＣＯＲＰＳは、遅延が大きくなる前には、ＳＬＩＰよりもずっと高い負荷にたえることができる。容易に分かるように、これらの曲線の微分は、高負荷の場合、ＲＲＧＳ及びＣＯＲＰＳのほうがかなり小さい。しかし、いずれのアルゴリズムも、中程度から軽い負荷ではオフセット遅延バジェットを有する。ＲＲＧＳの場合、これは、パイプライン法が使用されていることのみによるものである。ＣＯＲＰＳの場合、既に説明したように、追加遅延はコリジョン解決によるものである。しかし、ＣＯＲＰＳは、ＲＲＧＳに比べて２つの利点を有する。

（ｉ）ＳＭがどの出力ポートを選択するかについて選択の自由がある。

（ｉｉ）厳密に公平なスケジューラである。

ＳＬＩＰもまた公平なスケジューラであるが、そのコリジョン解決プロセスは、ＣＯＲＰＳのものとは全く異なる。

【０１１８】前述のように、ＣＯＲＰＳは、どの出力ポートにスケジューリングを試みるかについて完全な選択の自由を与える。すなわち、各ＶＯＱＭは、与えられたＶＯＱに代わって、スケジューリングされる出力ポートを自由に選択することができる。このことは、スケジューラ設計ストラテジの重要な部分であった。従って、多くのアルゴリズムが、ＣＯＲＰＳとともに、ＶＯＱ選択に使用可能である。これまで、そのようなアルゴリズムの1つ、すなわち、空でないＶＯＱのうちのランダム選択について説明した。他のＶＯＱ選択ストラテジの例も可能である。ＶＯＱ選択ストラテジは、協調的選択ストラテジおよび非協調的選択ストラテジという２つのクラスに分類することができる。

【０１１９】非協調的ＶＯＱ選択ストラテジは、ＶＯＱ選択決定が、他の入力ポートとは独立に、入力ポート（ＶＯＱＭ）ごとに行われるものである。ＣＯＲＰＳの分析に用いたランダム選択ストラテジはこのクラスに属する。

【０１２０】重み付き公平キューイング（ＷＦＱ：Weighted Fair Queuing）は、パケット交換研究文献において広く知られたサービスストラテジである（例えば、H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks", In Proceedings of IEEE, Vol.83, no.10, pp.1374-1396,Oct. 1995、参照）。その考え方は、所定の重みに従って、出力リンク容量に対して競合する複数のキューのサービスレートを規制するというものである。ＶＯＱ ＣＯＲＰＳスイッチにおいて、出力ポート帯域は、ある種の呼受付けコントローラによって複数のＶＯＱＭに分割することができる。その場合に、ＷＦＱを用いて、ＶＯＱキューの最大サービスレートが、与えられた出力ポートのＶＯＱＭ帯域分を超えないように強制することができる。

【０１２１】レート制御サービス（ＲＣＳ：Rate-Controlled Service）規律は、与えられたトラフィックフローが、ネットワークエントリポイントでいくつかのバースト性制約を満たすと仮定する（L. Georgiadis, R. Guerin, V. Pens, "EfficientNetwork QoS Provisioning Based on per Node Traffic Shaping", Proceedingsof INFOCOM96, vol.1, pp.102-110, 1996、参照）。これらの制約は一般に、ネットワークのエッジにおけるトラフィックシェーパによって強制される。さらに、トラフィックシェーパは、中間スイッチにも配置され、トラフィックが、ネットワーク内の各中間交換ポイントでそれらの制約に従うようにされる。トラフィックシェーパは一般に、リーキーバケットアルゴリズムによって実現される。J.Turner, "New Directions in Communications, or Which Way to the Information Age?", IEEE Communications Magazine, Vol.24, pp.8-15, 1986、には、そのようなアルゴリズムの１つが記載されている。基本的なリーキーバケットは、２つのキュー（１つはデータ用で、１つはトークンすなわちパーミット用）を有するシステムである。キュー上のデータパケットは、サービスを受けるためにはパーミットを必要とする。制限された個数のパーミットのみがパーミットキューに格納される。パーミットは、一定レートで生成される。この種のトラフィックシェーパは、ＶＯＱのうちのいずれがサービスを受けるかを規制するために使用可能である。適格なＶＯＱのうちからは、キュー選択に任意のアルゴリズムを用いることが可能である。

【０１２２】上記の２つのサービス規律は、パケットネットワークにおけるサービス品質（ＱｏＳ）のサポートに使用可能であり、それ自体、活発な研究分野である。このようなＱｏＳサポートストラテジは、非協調的なタイプのものになることが多い。それは、他のトラフィックストリームとは無関係に、ＶＯＱの予測されるサービス挙動を保証することになるからである。このクラスに属するアルゴリズムは、ビデオや音声ストリームのような、厳しいＱｏＳアプリケーションをサポートするスイッチで使用可能である。

【０１２３】協調的ＶＯＱ選択ストラテジは、ＶＯＱ選択がスイッチ内のＶＯＱのセット全体の状態に依存するような選択ストラテジである。このストラテジは一般に、各フローのサービスに集中するよりも、最大スループットのようなスイッチ全体のふるまいを良くすることを目標とする。従って、このようなストラテジをスイッチで使用するのは、ＱｏＳ要求条件に対する約束なしに、データトラフィックをサポートする場合である。

【０１２４】協調的ストラテジの場合、他のＶＯＱの状態のような追加情報をＣＯＲＰＳスケジューラに提供する必要がある。キューの状態に関する情報は常に「古い」ため、サービスストラテジは、古い情報に関してロバストでなければならない。

【0125】最大マッチング問題とは、与えられたグラフの辺のうちから、グラフの頂点の対をつなぐ辺で、対の総数を最大にするような辺のサブセットを求める問題である（Cormen, Leiserson and Rivest, "Introduction to Algorithms", McGraw-Hill, 1990、参照）。しかし、どの頂点も、つなぐ選択された辺を複数本有することはできない。あらゆるスロットで交換されるパケットの個数を最大にする場合、最大2部マッチング（MBM：Maximum Bipartite Matching）問題を解く必要がある（R. E. Tarjan, "Data Structures and Network Algorithms", Society for Industrial and Applied Mathematics, Pennsylvania, Nov. 1983、参照）。適当な計算量でMBMを解くアルゴリズムが利用可能である（J. E. Hopcroft, R. M. Karp, "An $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs", Society for Industrial and Applied Mathematics J. Comput., 2 (1973), pp.225-231、参照）。本発明では、VOQが空きであるか否かの状態情報は、通信チェインを通じて送られて、VOQMに渡される。ここで、MBMアルゴリズムは、次フレームのスロットにどのキューがサービスするかを決定する。興味深い点であるが、CORPSによれば、MBMアルゴリズムによって選択されないキューも、将来の予約を試みることが可能である。

【0126】最大重み2部マッチング（MWBM：Maximum Weight Bipartite Matching）問題は、上記のMBM問題と類似している。主な相違点は、前者では、重みがグラフの辺に関連づけられ、目的は、マッチングの辺の重みの総和を最大にする辺のセットを求めることであることである。他の研究者は、MWBMアルゴリズムを用いると、非一様トラフィックのもとでは、スループットに関してMBMストラテジよりパフォーマンスが優れていることを示している（N. McKeown, V. Anantharam, J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch", Proceedings of Infocom96, San Francisco, March 1996、参照）。考え方は、非一様トラフィックの場合を扱うために、VOQキューサイズを重みとして用いることである。

【0127】また、上記文献によれば、入力トラフィックが受付け可能である限り、MWBMアルゴリズムは安定である、すなわち、VOQキューは爆発しない。あらゆる出力ポートに対して、1個の出力ポートへの入力トラフィックレートの総和がその容量を超えない場合に、トラフィックが受付け可能であるという。この興味深い結果は、MWBNの安定性は、古い情報の存在下でも、すなわち、重みがいくつかの過去のタイムスロットのキューレベルに基づいていても、維持されるということである。この場合も、VOQのキューレベル情報はすべてのVOQMに渡され、出力ポートに対する要求がSMへと発行される前に、MWBMアルゴリズムが各モジュールで実行されるようにすることができる。

【0128】

【発明の効果】以上詳細に説明したように、本発明によれば、まず、時間軸をフレーム化して優先マトリクスを用いた予約を行うことにより、タイムスロット巡回順序を規則的なものとし、実装及び制御を容易にするという効果がある。また、RRGSとは異なり、スケジューリングの規則をSM個数の偶奇によって変える必要がなく、この点でも実装及び制御が簡単であるといえる。

【0129】さらに、本発明による繰越しラウンドロビンパイプラインスケジューラ（CORPS）によれば、クロスバ高速スイッチファブリックの入線間での公平なスケジューリングが可能となる。CORPSは、将来のスロットのパケットをスケジューリングすることにより、ラインごとスロットごとに1つのスケジューリング決定を行う。スケジューリングされるキューの選択は任意であるため、トラフィックのサービス品質をサポートすることに適している。CORPSは、出力ポート間の競合を公平に解決する。

【0130】さらに他の効果および変形を考えることは当業者には容易であって、本発明は、ここで説明した具体例に限定されない。特許請求の範囲に記載した本発明の構成の技術思想あるいは技術的範囲から離れることなく、さまざまな変形例を考えることが可能である。

【図面の簡単な説明】

【図1】集中VOQスケジューラを示す概略的ブロック図である。

【図2】（A）は並列方式の、（B）はラウンドロビン方式のアーキテクチャをそれぞれ示す分散スケジューラアーキテクチャの模式図である。

【図3】入力バッファ型スイッチアーキテクチャの説明図である。

【図4】入力ポート分散スケジューラの構成を例示する概略的ブロック図である。

【図5】4×4クロスバースイッチを用いた場合のRRGSによるパイプラインスケジューリング決定の一例を示すタイムチャートである。

【図6】図5を個々のSMのタイムスロット巡回順序に着目して表現したタイムチャートである。

【図7】本発明によるスケジューラの一実施形態におけるコリジョンを解決するのに用いられる優先マトリクスの一例（ポート数N＝4）を示す説明図である。

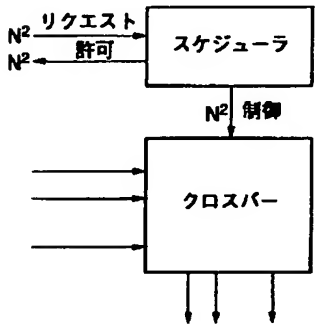
【図8】本発明によるスケジューラの一実施形態におけるパイプラインスケジューリング決定の一例（ポート数N＝4）を示す説明図である。

【図9】本発明によるスケジューラの一実施形態におけるコリジョンを解決するのに用いられる優先マトリクスの一例（ポート数N＝5）を示す説明図である。

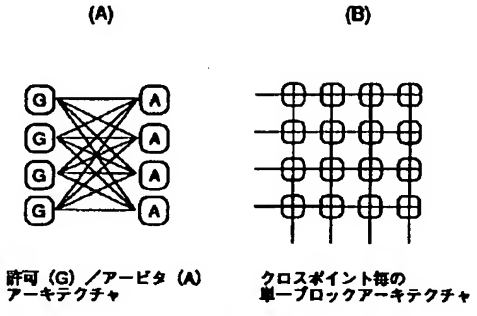
【図10】本発明によるスケジューラの一実施形態におけるパイプラインスケジューリング決定の一例（ポート数N＝5）を示す説明図である。

【図１１】本発明によるスケジューラの一実施形態におけるＳＭ間の繰越し動作を示す説明図である。
【図１２】本実施形態におけるＳメッセージのフォーマット図である。
【図１３】本実施形態におけるスケジューラモジュールのデータ構造体のフォーマット図である。
【図１４】本実施形態におけるＣＯＲＰＳスケジューリングアルゴリズムを示すフローチャートである。
【図１５】ＣＯＲＰＳ　ＶＯＱキューイングモデルを示す模式図である。
【図１６】システム負荷の関数としてパケット遅延を表すグラフである。
【図１７】ＣＯＲＰＳスケジューラを備えた１６×１６スイッチの相補的遅延分散を示すグラフである。
【図１８】ＣＯＲＰＳコントローラの一例を示すブロック図である。
【図１９】さまざまな競合スケジューラの、システム負荷に対する期待遅延を示すグラフである。

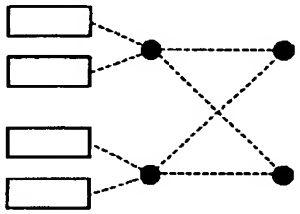
【図1】



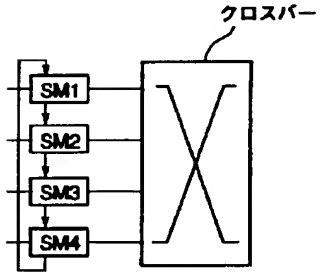
【図2】



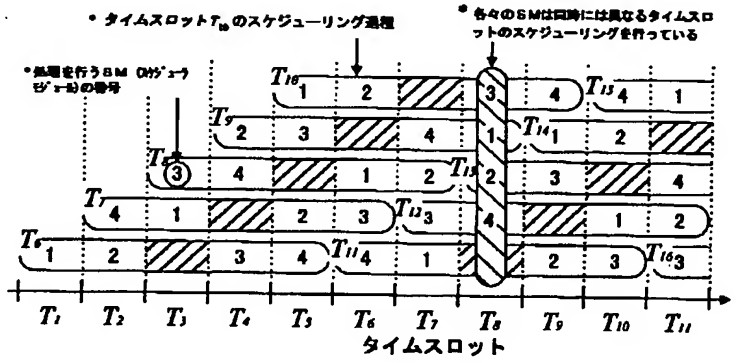
【図3】



【図4】



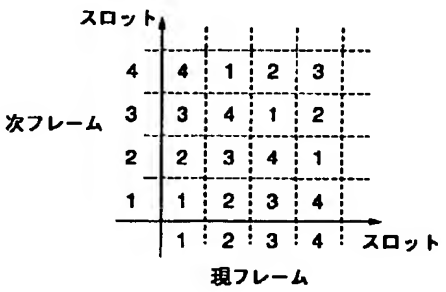
【図5】



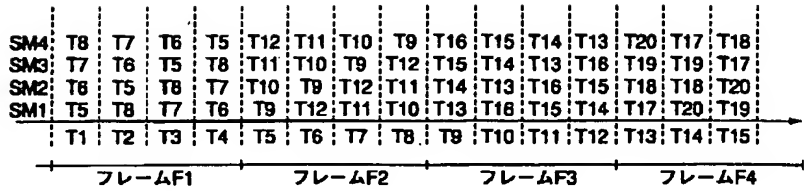
【図6】

タイムスロット / スケジューリングモジュール	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
SM1	T6	T4	T7	T5	T10	T8	T11	T9	T14	T12	T15
SM2	T3	T6	T4	T9	T7	T10	T8	T13	T11	T14	T12
SM3	T5	T3	T8	T6	T9	T7	T12	T10	T13	T11	T16
SM4	T2	T7	T5	T8	T6	T11	T9	T12	T10	T15	T13

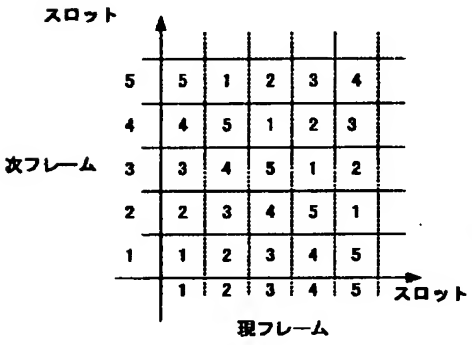
【図7】



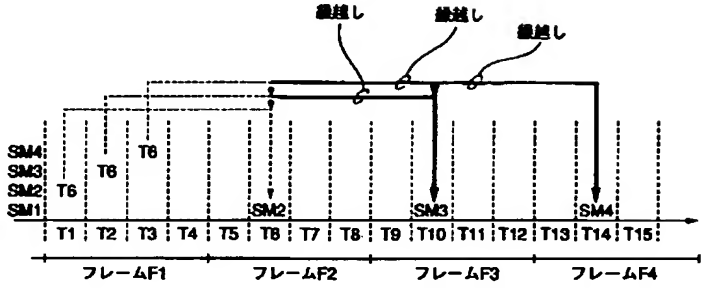
【図8】



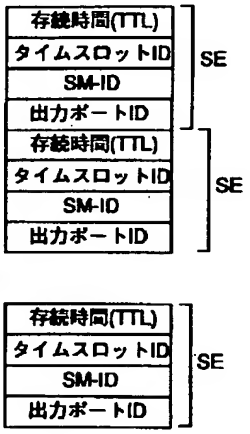
【図9】



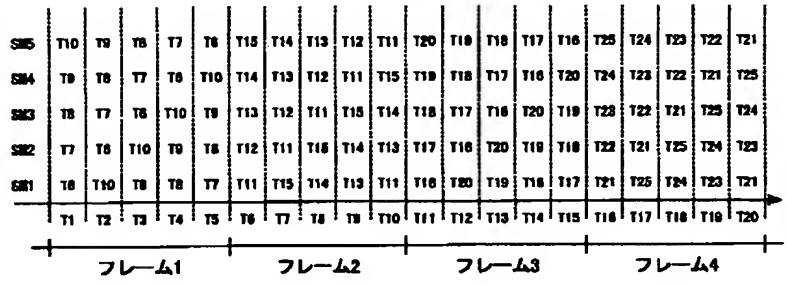
【図11】



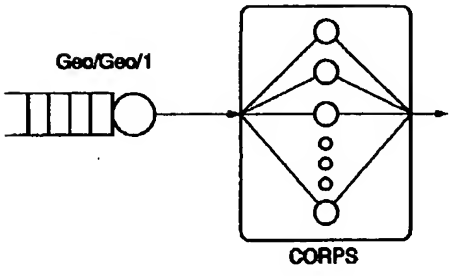
【図12】



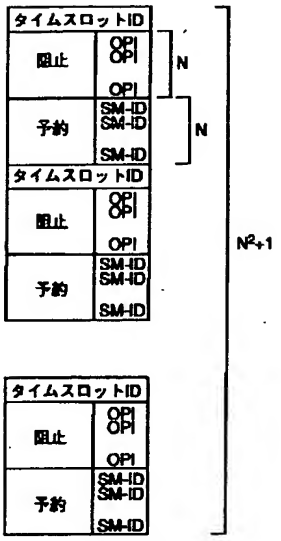
【図10】



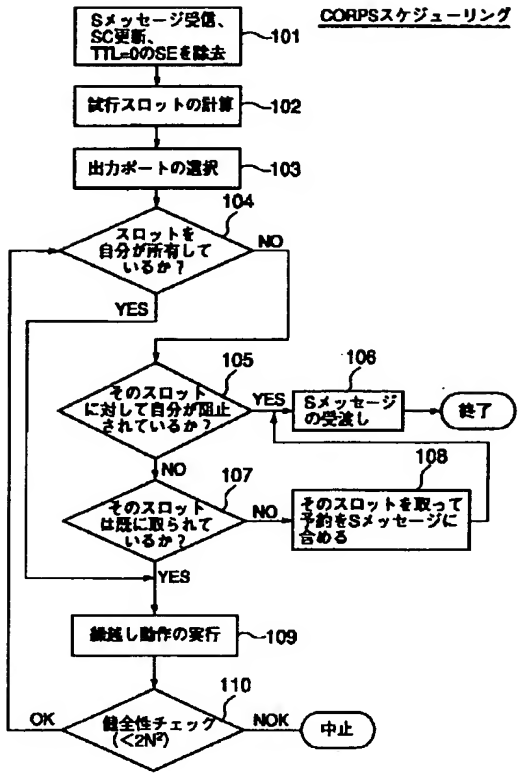
【図15】



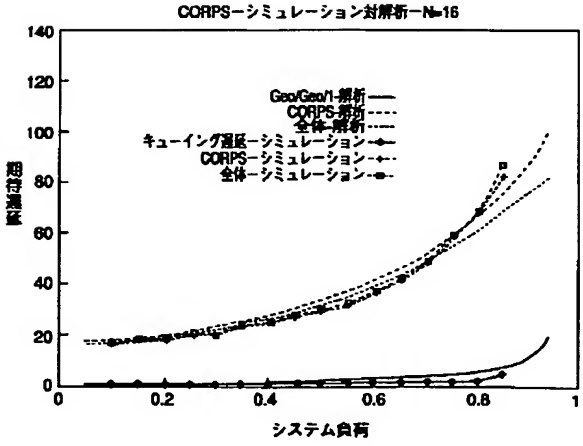
【図13】



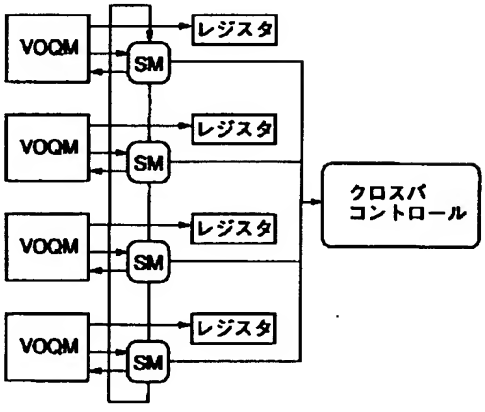
【図14】



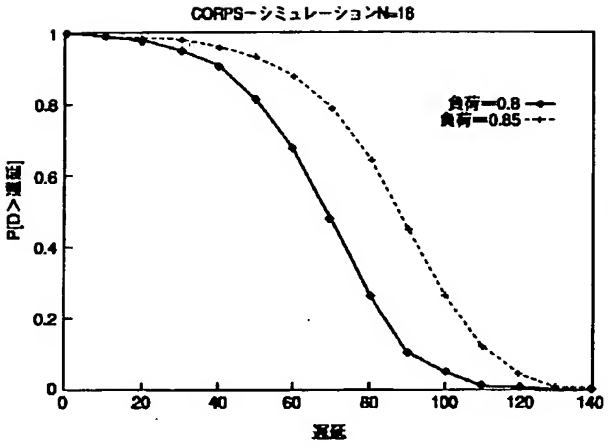
【図16】



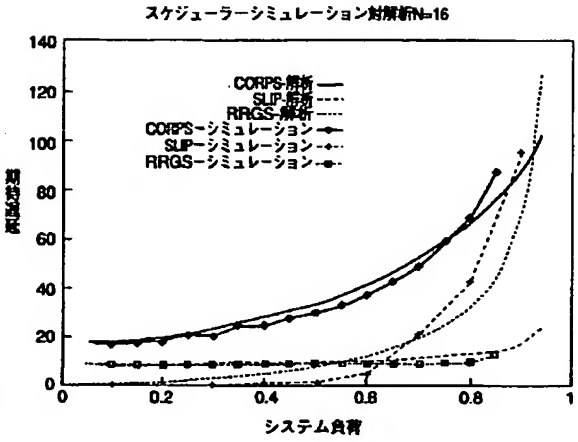
【図18】



【図17】


CORPSーシミュレーションN=16
負荷=0.8
負荷=0.85
P[D>遅延]
遅延
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100
120
140

【図19】


スケジューラーシミュレーション対解析N=16
CORPS-解析
SLIP-解析
RRGS-解析
CORPSーシミュレーション
SLIPーシミュレーション
RRGSーシミュレーション
システム負荷
0
20
40
60
80
100
120
140
0.2
0.4
0.6
0.8
1